L 12

```
   SSSSSSSSSSSSS  YYY          YYY      SSSSSSSSSSSSS
   SSSSSSSSSSSSS  YYY          YYY      SSSSSSSSSSSSS
   SSSSSSSSSSSSS  YYY          YYY      SSSSSSSSSSSSS
SSS               YYY          YYY   SSS
SSS               YYY          YYY   SSS
SSS               YYY          YYY   SSS
SSS                   YYY   YYY      SSS
SSS                   YYY   YYY      SSS
SSS                   YYY   YYY      SSS
   SSSSSSSSSS            YYY            SSSSSSSSSS
   SSSSSSSSSS            YYY            SSSSSSSSSS
   SSSSSSSSSS            YYY            SSSSSSSSSS
             SSS         YYY                      SSS
             SSS         YYY                      SSS
             SSS         YYY                      SSS
             SSS         YYY                      SSS
             SSS         YYY                      SSS
             SSS         YYY                      SSS
SSSSSSSSSSSSS            YYY         SSSSSSSSSSSSS
SSSSSSSSSSSSS            YYY         SSSSSSSSSSSSS
SSSSSSSSSSSSS            YYY         SSSSSSSSSSSSS
```

_$

Ps

YZ

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

FILEID**IMGDECODE

```
IIIIII    MM      MM    GGGGGGGG  DDDDDDDD   EEEEEEEEEE    CCCCCCCC    OOOOOO    DDDDDDDD    EEEEEEEEEE
IIIIII    MM      MM    GGGGGGGG  DDDDDDDD   EEEEEEEEEE    CCCCCCCC    OOOOOO    DDDDDDDD    EEEEEEEEEE
  II      MMMM  MMMM  GG          DD      DD EE          CC          OO      OO  DD      DD  EE
  II      MMMM  MMMM  GG          DD      DD EE          CC          OO      OO  DD      DD  EE
  II      MM  MM  MM  GG          DD      DD EE          CC          OO      OO  DD      DD  EE
  II      MM  MM  MM  GG          DD      DD EE          CC          OO      OO  DD      DD  EE
  II      MM      MM  GG          DD      DD EEEEEEEE    CC          OO      OO  DD      DD  EEEEEEEE
  II      MM      MM  GG          DD      DD EEEEEEEE    CC          OO      OO  DD      DD  EEEEEEEE
  II      MM      MM  GG  GGGGGG  DD      DD EE          CC          OO      OO  DD      DD  EE
  II      MM      MM  GG  GGGGGG  DD      DD EE          CC          OO      OO  DD      DD  EE
  II      MM      MM  GG      GG  DD      DD EE          CC          OO      OO  DD      DD  EE
  II      MM      MM  GG      GG  DD      DD EE          CC          OO      OO  DD      DD  EE
IIIIII    MM      MM    GGGGGG    DDDDDDDD   EEEEEEEEEE    CCCCCCCC    OOOOOO    DDDDDDDD    EEEEEEEEEE
IIIIII    MM      MM    GGGGGG    DDDDDDDD   EEEEEEEEEE    CCCCCCCC    OOOOOO    DDDDDDDD    EEEEEEEEEE


LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;  1     0001  0 %TITLE  'Get and Decode Image Header and Sections'
;  2     0002  0 MODULE IMG$DECODE (
;  3     0003  0                 LANGUAGE (BLISS32),
;  4     0004  0                 IDENT = 'V04-000'
;  5     0005  0                 ) =
;  6     0006  1 BEGIN
;  7     0007  1
;  8     0008  1 !
;  9     0009  1 !*****************************************************************************
; 10     0010  1 !*                                                                           *
; 11     0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 12     0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
; 13     0013  1 !*  ALL RIGHTS RESERVED.                                                      *
; 14     0014  1 !*                                                                           *
; 15     0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
; 16     0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
; 17     0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
; 18     0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
; 19     0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
; 20     0020  1 !*  TRANSFERRED.                                                              *
; 21     0021  1 !*                                                                           *
; 22     0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
; 23     0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
; 24     0024  1 !*  CORPORATION.                                                              *
; 25     0025  1 !*                                                                           *
; 26     0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
; 27     0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
; 28     0028  1 !*                                                                           *
; 29     0029  1 !*                                                                           *
; 30     0030  1 !*****************************************************************************
; 31     0031  1 !
; 32     0032  1
; 33     0033  1 !++
; 34     0034  1 ! FACILITY:     Exec, Shareable routines to decode image header and sections
; 35     0035  1 !
; 36     0036  1 ! ABSTRACT:
; 37     0037  1 !
; 38     0038  1 !       This module contains the routines to retrieve and decode
; 39     0039  1 !       an image header and the image section descriptors.
; 40     0040  1 !
; 41     0041  1 ! ENVIRONMENT:  VAX/VMS Operating System
; 42     0042  1 !
; 43     0043  1 ! AUTHOR:       Bob Grosso,     CREATION DATE:  16-Mar-1983
; 44     0044  1 !
; 45     0045  1 ! MODIFIED BY:
; 46     0046  1 !
; 47     0047  1 !       V03-010 MSH0051         Michael S. Harvey       20-May-1984
; 48     0048  1 !               Convert old format image name string to new format.
; 49     0049  1 !
; 50     0050  1 !       V03-009 MSH0043         Michael S. Harvey       8-May-1984
; 51     0051  1 !               When converting x-linker image headers into a modern
; 52     0052  1 !               form, update the image IDs correspondingly.
; 53     0053  1 !
; 54     0054  1 !       V03-008 MSH0041         Michael S. Harvey       2-May-1984
; 55     0055  1 !               Add some beef to the bounds checking code to ensure that
; 56     0056  1 !               only valid images are run. These checks filter obviously
; 57     0057  1 !               bad image headers and images with bad ISD lists.
```

```
;    58          0058  1 ;
;    59          0059  1 ;       V03-007 LJK0269         Lawrence J. Kenah       31-Mar-1984
;    60          0060  1 ;               Miscellaneous cleanup.
;    61          0061  1 ;               Do not perform consistence checks on TYPE 2 images. They are
;    62          0062  1 ;               not necessarily produced by the linker.
;    63          0063  1 ;               Make sure that a primitive length check is performed on the
;    64          0064  1 ;               IHD and ISD sizes before the buffer is copied.
;    65          0065  1 ;
;    66          0066  1 ;       V03-006 LJA0110         Laurie J. Anderson       6-Feb-1984
;    67          0067  1 ;               Change the error messages returned from the image decode
;    68          0068  1 ;               routines to be something more intelligent than "bad hdr".
;    69          0069  1 ;
;    70          0070  1 ;       V03-005 WMC0001         Wayne Cardoza           24-Jan-1984
;    71          0071  1 ;               Add support for cross-linker and V3 FT1 images.
;    72          0072  1 ;
;    73          0073  1 ;       V03-004 LJK0243         Lawrence J. Kenah       23-Aug-1983
;    74          0074  1 ;               Return IHD$Q_PRIVREQS of all privileges for old images,
;    75          0075  1 ;               ones that do not contain a SYSVER field.
;    76          0076  1 ;
;    77          0077  1 ;       V03-003 LJK0234         Lawrence J. Kenah       26-Jul-1983
;    78          0078  1 ;               Fix code that transforms old image header into latest
;    79          0079  1 ;               form of image header.
;    80          0080  1 ;
;    81          0081  1 ;       V03-002 LJK0229         Lawrence J. Kenah       12-Jul-1983
;    82          0082  1 ;               Treat the alias and offset as words. Treat the ISD
;    83          0083  1 ;               size as a signed word.
;    84          0084  1 ;
;    85          0085  1 ;       V03-001 LJK0223         Lawrence J. Kenah       6-Jul-1983
;    86          0086  1 ;               Make IHD and ISD sizes into words so that the comparisons
;    87          0087  1 ;               are made correctly.
;    88          0088  1 ;--
```

```
;    90          0089  1 %SBTTL  'Definitions'
;    91          0090  1
;    92          0091  1 !
;    93          0092  1 ! INCLUDE FILES:
;    94          0093  1 !
;    95          0094  1
;    96          0095  1 LIBRARY 'SYS$LIBRARY:LIB.L32';                  ! Define system data structures
;    97          0096  1
;    98          0097  1 REQUIRE 'LIB$:IMGMSGDEF.R32';                   ! Get status code definitions
;    99          0183  1
;   100          0184  1 !
;   101          0185  1 ! PSECT DECLARATIONS:
;   102          0186  1 !
;   103          0187  1
;   104          0188  1 PSECT
;   105          0189  1         CODE    = YF$$SYSIMGACT (WRITE),
;   106          0190  1         PLIT    = YF$$SYSIMGACT (WRITE, EXECUTE);
;   107          0191  1
;   108          0192  1 !
;   109          0193  1 ! LITERALS
;   110          0194  1 !
;   111          0195  1
;   112          0196  1 LITERAL
;   113          0197  1         TRUE = 1,
;   114          0198  1         FALSE = 0,
;   115          0199  1
;   116          0200  1         IMG$C_BLOCKSIZ = 512;
;   117          0201  1
;   118          0202  1 !
;   119          0203  1 ! EXTERNAL REFERENCES:
;   120          0204  1 !
;   121          0205  1
;   122          0206  1 EXTERNAL LITERAL
;   123          0207  1
;   124          0208  1         EXE$C_SYSEFN : UNSIGNED (6);    ! System event flag for QIO Wait read
;   125          0209  1
;   126          0210  1 !
;   127          0211  1 ! FORWARD ROUTINE REFERENCES
;   128          0212  1 !
;   129          0213  1
;   130          0214  1 FORWARD ROUTINE
;   131          0215  1         CONVERT_XLINK;
;   132          0216  1
;   133          0217  1 !
;   134          0218  1 ! Define VMS block structures
;   135          0219  1 !
;   136          0220  1 STRUCTURE
;   137          0221  1         BBLOCK [O, P, S, E; N] =
;   138          0222  1                 [N]
;   139          0223  1                 (BBLOCK + O) <P, S, E>;
;   140          0224  1
```

```
;  142     0225  1 %SBTTL  'IMG$DECODE_IHD  Get Image Header'
;  143     0226  1 GLOBAL ROUTINE IMG$DECODE_IHD
;  144     0227  1         ( CHAN, BLK_BUFADR, IHD_BUFADR, VBN_ADR, OFFSET_ADR, HDRVER_ADR, ALIAS_ADR ) =
;  145     0228  1
;  146     0229  1 !++
;  147     0230  1 ! FUNCTIONAL DESCRIPTION:
;  148     0231  1 !
;  149     0232  1 ! FORMAL PARAMETERS:
;  150     0233  1 !
;  151     0234  1 !       Chan            Channel on which image file is open
;  152     0235  1 !       Blk_bufadr      Address of buffer to contain 1st block of image
;  153     0236  1 !       Ihd_bufadr      Address of buffer to contain decoded IHD
;  154     0237  1 !       VBN_adr         Address of VBN to be set to 1
;  155     0238  1 !       Offset_adr      Address of Offset in which to return offset to 1st ISD
;  156     0239  1 !
;  157     0240  1 ! IMPLICIT INPUTS:
;  158     0241  1 !       NONE
;  159     0242  1 !
;  160     0243  1 ! IMPLICIT OUTPUTS:
;  161     0244  1 !       NONE
;  162     0245  1 !
;  163     0246  1 ! ROUTINE VALUE:
;  164     0247  1 ! COMPLETION CODES:
;  165     0248  1 !       NONE
;  166     0249  1 !
;  167     0250  1 ! SIDE EFFECTS:
;  168     0251  1 !       NONE
;  169     0252  1 !
;  170     0253  1 !--
;  171     0254  1
;  172     0255  2 BEGIN
;  173     0256  2
;  174     0257  2 LITERAL
;  175     0258  2     IHDMAXSIZ = IHD$C_LENGTH +  ! Maximum length for fixed portion of header
;  176     0259  2                 IHA$C_LENGTH +
;  177     0260  2                 IHS$C_LENGTH +
;  178     0261  2                 IHI$C_LENGTH +
;  179     0262  2                 IHP$C_LENGTH ,
;  180     0263  2     IHI_S_IMGNAM = 16;          ! Length of image name string prior to VMS V4
;  181     0264  2
;  182     0265  2 LOCAL
;  183     0266  2     B_IHD : REF BBLOCK,         ! Buffer IHD
;  184     0267  2     IRD : REF BBLOCK,
;  185     0268  2     IOSB : BBLOCK [8],          ! Quadword IO status block
;  186     0269  2     HDR_INSERT,
;  187     0270  2     IHI_INSERT,
;  188     0271  2     OFF2 : WORD,
;  189     0272  2     SIZE,
;  190     0273  2     STATUS;                     ! Status
;  191     0274  2
;  192     0275  2 BIND
;  193     0276  2     V4_MAJORID = UPLIT (%ASCII'02'),            ! Major ID for VMS V4 images
;  194     0277  2     V4_MINORID = UPLIT (%ASCII'05'),            ! Minor ID for VMS V4 images
;  195     0278  2     HEADER_VERSION = .HDRVER_ADR        : WORD,
;  196     0279  2     LAST_WORD = .ALIAS_ADR              : SIGNED WORD,
;  197     0280  2     OFFSET = .OFFSET_ADR                : WORD,
;  198     0281  2     VBN = .VBN_ADR;
```

```
;  199          0282  2
;  200          0283  2 VBN = 1;                          ! Read from first block
;  201          0284  2 SIZE = IMG$C_BLOCKSIZ;           ! Read one block
;  202          0285  2
;  203          0286  2
;  204          0287  2 !
;  205          0288  2 !       Read first block
;  206          0289  2 !
;  207        P 0290  2 STATUS = $QIOW (
;  208        P 0291  2                 EFN = EXE$C_SYSEFN,     ! Event flag
;  209        P 0292  2                 CHAN = .CHAN,           ! Channel
;  210        P 0293  2                 FUNC = IO$_READVBLK,    ! Read a virtual block
;  211        P 0294  2                 IOSB = IOSB,            ! I/O status block
;  212        P 0295  2                 P1 = .BLK_BUFADR,       ! Buffer to read in to
;  213        P 0296  2                 P2 = .SIZE,             ! Number of bytes to read
;  214        P 0297  2                 P3 = .VBN               ! Virtual block number to read
;  215          0298  2                );
;  216          0299  2
;  217          0300  2 IF .STATUS
;  218          0301  2 THEN
;  219          0302  2     STATUS = .IOSB [0,0,16,0];          ! Pick up final status
;  220          0303  2 IF NOT .STATUS
;  221          0304  2 THEN
;  222          0305  2     RETURN .STATUS;
;  223          0306  2
;  224          0307  2 B_IHD = .BLK_BUFADR;                  ! Image header
;  225          0308  2 LAST_WORD = .B_IHD [IHD$W_ALIAS];      ! Contents of last word of header block
;  226          0309  2
;  227          0310  2 !
;  228          0311  2 !       Process the image based upon which type of image it is. Screen
;  229          0312  2 !       out obvious image pretenders.
;  230          0313  2 !
;  231          0314  2 CASE .LAST_WORD
;  232          0315  2 FROM IHD$C_MINCODE TO IHD$C_MAXCODE OF
;  233          0316  2     SET
;  234          0317  2
;  235          0318  2     [IHD$C_RSX, IHD$C_BPA, IHD$C_ALIAS] :
;  236          0319  2
;  237          0320  3         BEGIN
;  238          0321  3         CH$MOVE (IMG$C_BLOCKSIZ, .B_IHD, .IHD_BUFADR);  ! Copy image header to buffer
;  239          0322  3         HEADER_VERSION = 0;
;  240          0323  3         RETURN SS$_NORMAL;
;  241          0324  2         END;
;  242          0325  2
;  243          0326  2     [IHD$C_NATIVE, IHD$C_CLI] :
;  244          0327  2
;  245          0328  3         BEGIN
;  246          0329  3         IF .B_IHD [IHD$W_MAJORID] EQL IHX$K_MAJORID     ! If Cross linker format
;  247          0330  3         THEN
;  248          0331  4             BEGIN
;  249          0332  4             HEADER_VERSION = IHD$C_GEN_XLNKR;
;  250          0333  4             STATUS = CONVERT_XLINK (.BLK_BUFADR, .IHD_BUFADR);
;  251          0334  4             OFFSET = .B_IHD [IHD$W_SIZE];
;  252          0335  4             RETURN .STATUS;
;  253          0336  3             END;
;  254          0337  3
;  255          0338  3         !
```

```
;  256     0339  3          ! Check for a reasonable header record size and set of offsets.
;  257     0340  3          ! Simply verify that the offsets and the regions they point to
;  258     0341  3          ! fall within the image header record.
;  259     0342  3          !
;  260     0343  3          OFFSET = .B_IHD [IHD$W_SIZE];
;  261     0344  4          IF (.OFFSET LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
;  262     0345  3             OR
;  263     0346  4             (.OFFSET GTRU IHDMAXSIZ)
;  264     0347  4          THEN
;  265     0348  3              RETURN IMG$_IMG_SIZ;
;  266     0349  3
;  267     0350  3          !       Verify range of activation data offset
;  268     0351  3
;  269     0352  3          OFF2 = .B_IHD [IHD$W_ACTIVOFF];
;  270     0353  4          IF (.OFF2 LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
;  271     0354  3             OR
;  272     0355  4             (.OFF2 + IHA$C_LENGTH GTRU IHDMAXSIZ)
;  273     0356  3          THEN
;  274     0357  3              RETURN IMG$_BADOFFSET;
;  275     0358  3
;  276     0359  3          !       Verify range of debug and global symbol table offset
;  277     0360  3
;  278     0361  3          IF .B_IHD [IHD$W_SYMDBGOFF] NEQ 0
;  279     0362  3          THEN
;  280     0363  4              BEGIN
;  281     0364  4              OFF2 = .B_IHD [IHD$W_SYMDBGOFF];
;  282     0365  5              IF (.OFF2 LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
;  283     0366  4                 OR
;  284     0367  5                 (.OFF2 + IHS$C_LENGTH GTRU IHDMAXSIZ)
;  285     0368  4              THEN
;  286     0369  4                  RETURN IMG$_BADOFFSET;
;  287     0370  3              END;
;  288     0371  3
;  289     0372  3          !       Verify range of image ID data offset
;  290     0373  3
;  291     0374  3          OFF2 = .B_IHD [IHD$W_IMGIDOFF];
;  292     0375  4          IF (.OFF2 LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
;  293     0376  3             OR
;  294     0377  4             (.OFF2 + IHI$C_LENGTH GTRU IHDMAXSIZ)
;  295     0378  3          THEN
;  296     0379  3              RETURN IMG$_BADOFFSET;
;  297     0380  3
;  298     0381  3          !       Verify range of patch data offset
;  299     0382  3
;  300     0383  3          IF .B_IHD [IHD$W_PATCHOFF] NEQ 0
;  301     0384  3          THEN
;  302     0385  4              BEGIN
;  303     0386  4              OFF2 = .B_IHD [IHD$W_PATCHOFF];
;  304     0387  5              IF (.OFF2 LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
;  305     0388  4                 OR
;  306     0389  5                 (.OFF2 + IHP$C_LENGTH GTRU IHDMAXSIZ)
;  307     0390  4              THEN
;  308     0391  4                  RETURN IMG$_BADOFFSET;
;  309     0392  3              END;
;  310     0393  3
;  311     0394  3          !
;  312     0395  3          ! Copy image header to buffer
```

```
;  313     0396  3          !
;  314     0397  3          CH$MOVE (.B_IHD [IHD$W_SIZE], .B_IHD, .IHD_BUFADR);
;  315     0398  3
;  316     0399  3          HDR_INSERT = 0;                         ! Length by which header will be pried open
;  317     0400  3          HEADER_VERSION = IHD$C_GEN_FIXUP;       ! Default to most current
;  318     0401  3          IHD = .IHD_BUFADR;
;  319     0402  3
;  320     0403  3          !
;  321     0404  3          ! Calculate the degree by which the fixed portion of this header
;  322     0405  3          ! differs from the current format of the fixed part of an image header.
;  323     0406  3          ! Then, expand the fixed portion of the header by the required amount,
;  324     0407  3          ! thus converting it to the current format as if the image had been
;  325     0408  3          ! relinked.
;  326     0409  3          !
;  327     0410  3          IF $BYTEOFFSET (IHD$L_LNKFLAGS) GEQ .IHD [IHD$W_ACTIVOFF]
;  328     0411  3          THEN                                    ! Link flags were not present
;  329     0412  4              BEGIN                               ! so insert a longword
;  330     0413  4              HDR_INSERT = 4;
;  331     0414  4              HEADER_VERSION = IHD$C_GEN_NATIVE;
;  332     0415  3              END;
;  333     0416  3
;  334     0417  3          IF $BYTEOFFSET (IHD$L_SYSVER) GEQ .IHD [IHD$W_ACTIVOFF]
;  335     0418  3          THEN                                    ! System version and Ident were not pressent
;  336     0419  4              BEGIN                               ! so insert two blank longwords
;  337     0420  4
;  338     0421  4              BIND
;  339     0422  4                  PRIVILEGE_MASK = IHD [IHD$Q_PRIVREQS]   : VECTOR [2];
;  340     0423  4              .
;  341     0424  4              HDR_INSERT = .HDR_INSERT + 8;
;  342     0425  4              HEADER_VERSION = IHD$C_GEN_LNKFLG;
;  343     0426  4              PRIVILEGE_MASK [0] = -1;            ! Insure that image privilege mask
;  344     0427  4              PRIVILEGE_MASK [1] = -1;            !  indicates that all privileges are set
;  345     0428  3              END;
;  346     0429  3
;  347     0430  3          IF $BYTEOFFSET (IHD$L_IAFVA) GEQ .IHD [IHD$W_ACTIVOFF]
;  348     0431  3          THEN                                    ! Relative virtual address of fixup vector
;  349     0432  4              BEGIN                               ! not present so insert a blank longword
;  350     0433  4              HDR_INSERT = .HDR_INSERT + 4;
;  351     0434  4              HEADER_VERSION = IHD$C_GEN_SYSVER;
;  352     0435  3              END;
;  353     0436  3
;  354     0437  3
;  355     0438  3          IF .HDR_INSERT NEQ 0
;  356     0439  3          THEN                                    ! Shift non-fixed portion of image
;  357     0440  4              BEGIN                               ! to insert missing part of fixed section
;  358     0441  4
;  359     0442  4              CH$MOVE (
;  360     0443  4                  (.IHD [IHD$W_SIZE] - .IHD [IHD$W_ACTIVOFF]),    ! Shift the portion beginning at the
;  361     0444  4                  (.IHD + .IHD [IHD$W_ACTIVOFF]),                 !  point located by the first offset
;  362     0445  4                  (.IHD + .IHD [IHD$W_ACTIVOFF] + .HDR_INSERT));  !  by the amount to be inserted
;  363     0446  4
;  364     0447  4              CH$FILL (0, .HDR_INSERT,                ! Fill the space created for the insert
;  365     0448  4                  .IHD + .IHD [IHD$W_ACTIVOFF]);
;  366     0449  3              END;
;  367     0450  3
;  368     0451  3          !
;  369     0452  3          ! Determine the extent that the image ident area differs in size from
```

```
;  370     0453  3          ! the current format. Expand the image ident area by the required
;  371     0454  3          ! amount, thus converting to the current format without relinking.
;  372     0455  3          !
;  373     0456  3          IHI_INSERT = 0;                         ! Assume no conversion required
;  374     0457  4          IF (.IHD [IHD$W_MAJORID] LSSU .V4_MAJORID)
;  375     0458  3             OR
;  376     0459  4             (
;  377     0460  5              (.IHD [IHD$W_MAJORID] EQL .V4_MAJORID)
;  378     0461  4              AND
;  379     0462  5              (.IHD [IHD$W_MINORID] LSSU .V4_MINORID)
;  380     0463  4             )
;  381     0464  3          THEN
;  382     0465  3              !
;  383     0466  3              ! The image name string grew between VMS V3 and V4. Split the
;  384     0467  3              ! image ident area after the old image name string and expand
;  385     0468  3              ! the string to the current maximum size, zero filled.
;  386     0469  3              !
;  387     0470  4              BEGIN
;  388     0471  4              IHI_INSERT = IHI$S_IMGNAM - IHI_S_IMGNAM; ! Calculate size difference
;  389     0472  4              CH$MOVE (
;  390     0473  4                  (.IHD [IHD$W_SIZE] - (.IHD [IHD$W_IMGIDOFF] + IHI_S_IMGNAM)),
;  391     0474  4                  (.IHD + .IHD [IHD$W_IMGIDOFF] + IHI_S_IMGNAM),
;  392     0475  4                  (.IHD + .IHD [IHD$W_IMGIDOFF] + IHI$S_IMGNAM) );
;  393     0476  4              CH$FILL (0, .IHI_INSERT,
;  394     0477  4                  (.IHD + .IHD [IHD$W_IMGIDOFF] + IHI_S_IMGNAM) );
;  395     0478  3              END;
;  396     0479  3          !
;  397     0480  3          !   Correct all the offsets to compensate for the insertion(s). Note that two of
;  398     0481  3          !   the offsets locate optional parts of the image header and are only updated
;  399     0482  3          !   if the associated areas are present in the image (offsets are nonzero).
;  400     0483  3          !
;  401     0484  4              IF (.HDR_INSERT NEQ 0)
;  402     0485  3                 OR
;  403     0486  4                 (.IHI_INSERT NEQ 0)
;  404     0487  3              THEN
;  405     0488  4                  BEGIN
;  406     0489  4                  IHD [IHD$W_SIZE]      = .IHD [IHD$W_SIZE]      + .HDR_INSERT + .IHI_INSERT;
;  407     0490  4                  IHD [IHD$W_ACTIVOFF] = .IHD [IHD$W_ACTIVOFF] + .HDR_INSERT;
;  408     0491  4                  IHD [IHD$W_IMGIDOFF] = .IHD [IHD$W_IMGIDOFF] + .HDR_INSERT;
;  409     0492  4
;  410     0493  4                  IF .IHD [IHD$W_SYMDBGOFF] NEQU 0
;  411     0494  4                  THEN
;  412     0495  4                      IHD [IHD$W_SYMDBGOFF] = .IHD [IHD$W_SYMDBGOFF] + .HDR_INSERT;
;  413     0496  4
;  414     0497  4                  IF .IHD [IHD$W_PATCHOFF] NEQU 0
;  415     0498  4                  THEN
;  416     0499  4                      IHD [IHD$W_PATCHOFF]  = .IHD [IHD$W_PATCHOFF]  + .HDR_INSERT + .IHI_INSERT;
;  417     0500  4
;  418     0501  3                  END;
;  419     0502  3
;  420     0503  3              RETURN SS$_NORMAL;
;  421     0504  2              END;
;  422     0505  2
;  423     0506  2          [INRANGE,OUTRANGE] :
;  424     0507  2
;  425     0508  2              RETURN IMG$_BADHDR;             ! Unrecognizable or unsupported image type
;  426     0509  2
```

```
;  427          0510  2     TES;                              ! CASE of image types
;  428          0511  2
;  429          0512  1 END;                                  ! IMG$DECODE_IHD routine

                                                    .TITLE  IMG$DECODE Get and Decode Image Header and Sect
;                                                                   ions
                                                    .IDENT  \V04-000\

                                                    .PSECT  YF$$SYSIMGACT,2

                                    00  00  32  30  00000 P.AAA:  .ASCII  \02\<0><0>                                  ;
                                    00  00  35  30  00004 P.AAB:  .ASCII  \05\<0><0>                                  ;

                                                V4_MAJORID=         P.AAA
                                                V4_MINORID=         P.AAB
                                                    .EXTRN  EXE$C_SYSEFN, SYS$QIOW

                                          07FC 00000            .ENTRY  IMG$DECODE_IHD, Save R2,R3,R4,R5,R6,R7,R8,- ; 0226
                                                                        R9,R10                                      ;
                                       5E         08  C2 00002          SUBL2   #8, SP                              ;
                                       58    18   AC  D0 00005          MOVL    HDRVER_ADR, R8                      ; 0278
                                 10    BC         01  D0 00009          MOVL    #1, @VBN_ADR                        ; 0283
                                       50  0200   8F  3C 0000D          MOVZWL  #512, SIZE                          ; 0284
                                                  7E  7C 00012          CLRQ    -(SP)                               ; 0298
                                                  7E  D4 00014          CLRL    -(SP)                               ;
                                             10   BC  DD 00016          PUSHL   @VBN_ADR                            ;
                                                  50  DD 00019          PUSHL   SIZE                                ;
                                             08   AC  DD 0001B          PUSHL   BLK_BUFADR                          ;
                                                  7E  7C 0001E          CLRQ    -(SP)                               ;
                                             20   AE  9F 00020          PUSHAB  IOSB                                ;
                                                  31  DD 00023          PUSHL   #49                                 ;
                                             04   AC  DD 00025          PUSHL   CHAN                                ;
                                          7E         00G 9A 00028          MOVZBL  S^EXE$C_SYSEFN, -(SP)               ;
                                 00000000G  00         0C  FB 0002B          CALLS   #12, SYS$QIOW                   ;
                                          57         50  D0 00032          MOVL    R0, STATUS                          ;
                                          50         57  E9 00035          BLBC    STATUS, 4$                          ; 0300
                                          57         6E  3C 00038          MOVZWL  IOSB, STATUS                        ; 0302
                                          4A         57  E9 0003B          BLBC    STATUS, 4$                          ; 0303
                                          56    08   AC  D0 0003E          MOVL    BLK_BUFADR, B_IHD                   ; 0307
                                       1C    BC   01FE  C6  B0 00042          MOVW    510(B_IHD), @ALIAS_ADR              ; 0308
                                 04    FFFF  8F   1C    BC  AF 00048          CASEW   @ALIAS_ADR, #-1, #4                 ; 0328
0012                   0012                0012          001E     0004F 1$:      .WORD   3$-1$,-                     ;
                                                         001E     00057                 2$-1$,-                     ;
                                                                                        2$-1$,-                     ;
                                                                                        2$-1$,-                     ;
                                                                                        3$-1$                       ;
                                          50 084D8C84   8F  D0 00059          MOVL    #139299972, R0                  ; 0508
                                                        04  00060          RET                                     ;
                                 0C   BC         66   0200  8F  28 00061 2$:       MOVC3   #512, (B_IHD), @IHD_BUFADR  ; 0321
                                                        68  B4 00068          CLRW    (R8)                            ; 0322
                                                                 017A 31 0006A          BRW     19$                 ; 0328
                                       3130   8F        0C  A6  B1 0006D 3$:       CMPW    12(B_IHD), #12592           ; 0329
                                                        17  12 00073          BNEQ    5$                              ;
                                          68            01  B0 00075          MOVW    #1, (R8)                        ; 0332
                                          7E       08   AC  7D 00078          MOVQ    BLK_BUFADR, -(SP)               ; 0333
                                       0000V  CF         02  FB 0007C          CALLS   #2, CONVERT_XLINK               ;
```

```
                    57              50  D0 00081        MOVL    R0, STATUS                              :
               14   BC              66  B0 00084        MOVW    (B_IHD), @OFFSET_ADR                    : 0334
                    50              57  D0 00088 4$:    MOVL    STATUS, R0                              : 0335
                                        04 0008B        RET                                             :
               14   BC              66  B0 0008C 5$:    MOVW    (B_IHD), @OFFSET_ADR                    : 0343
                    20        14    BC  B1 00090        CMPW    @OFFSET_ADR, #32                        : 0344
                                    08  1F 00094        BLSSU   6$                                      :
             00D4   8F        14    BC  B1 00096        CMPW    @OFFSET_ADR, #212                       : 0346
                                    08  1B 0009C        BLEQU   7$                                      :
             50 084D8CA4            8F  D0 0009E 6$:    MOVL    #139300004, R0                          : 0348
                                        04 000A5        RET                                             :
                    50        02    A6  B0 000A6 7$:    MOVW    2(B_IHD), OFF2                          : 0352
                    20              50  B1 000AA        CMPW    OFF2, #32                               : 0353
                                    62  1F 000AD        BLSSU   9$                                      :
                    51              50  3C 000AF        MOVZWL  OFF2, R1                                : 0355
                    51              14  C0 000B2        ADDL2   #20, R1                                 :
         000000D4   8F              51  D1 000B5        CMPL    R1, #212                                :
                                    53  1A 000BC        BGTRU   9$                                      :
                              04    A6  B5 000BE        TSTW    4(B_IHD)                                : 0361
                                    18  13 000C1        BEQL    8$                                      :
                    50        04    A6  B0 000C3        MOVW    4(B_IHD), OFF2                          : 0364
                    20              50  B1 000C7        CMPW    OFF2, #32                               : 0365
                                    45  1F 000CA        BLSSU   9$                                      :
                    51              50  3C 000CC        MOVZWL  OFF2, R1                                : 0367
                    51              14  C0 000CF        ADDL2   #20, R1                                 :
         000000D4   8F              51  D1 000D2        CMPL    R1, #212                                :
                                    36  1A 000D9        BGTRU   9$                                      :
                    50        06    A6  B0 000DB 8$:    MOVW    6(B_IHD), OFF2                          : 0374
                    20              50  B1 000DF        CMPW    OFF2, #32                               : 0375
                                    2D  1F 000E2        BLSSU   9$                                      :
                    51              50  3C 000E4        MOVZWL  OFF2, R1                                : 0377
                    51        50    A1  9E 000E7        MOVAB   80(R1), R1                              :
         000000D4   8F              51  D1 000EB        CMPL    R1, #212                                :
                                    1D  1A 000F2        BGTRU   9$                                      :
                              08    A6  B5 000F4        TSTW    8(B_IHD)                                : 0383
                                    20  13 000F7        BEQL    10$                                     :
                    50        08    A6  B0 000F9        MOVW    8(B_IHD), OFF2                          : 0386
                    20              50  B1 000FD        CMPW    OFF2, #32                               : 0387
                                    0F  1F 00100        BLSSU   9$                                      :
                    50              50  3C 00102        MOVZWL  OFF2, R0                                : 0389
                    50              2C  C0 00105        ADDL2   #44, R0                                 :
         000000D4   8F              50  D1 00108        CMPL    R0, #212                                :
                                    08  1B 0010F        BLEQU   10$                                     :
             50 084D8C94            8F  D0 00111 9$:    MOVL    #139299988, R0                          : 0391
                                        04 00118        RET                                             :
  0C    BC          66              66  28 00119 10$:   MOVC3   (B_IHD), (B_IHD), @IHD_BUFADR           : 0397
                                    59  D4 0011E        CLRL    HDR_INSERT                              : 0399
                    68              05  B0 00120        MOVW    #5, (R8)                                : 0400
                    56        0C    AC  D0 00123        MOVL    IHD_BUFADR, IHD                         : 0401
                    20        02    A6  B1 00127        CMPW    2(IHD), #32                             : 0410
                                    06  1A 0012B        BGTRU   11$                                     :
                    59              04  D0 0012D        MOVL    #4, HDR_INSERT                          : 0413
                    68              02  B0 00130        MOVW    #2, (R8)                                : 0414
                    28        02    A6  B1 00133 11$:   CMPW    2(IHD), #40                             : 0417
                                    11  1A 00137        BGTRU   12$                                     :
                    50        14    A6  9E 00139        MOVAB   20(IHD), R0                             : 0422
                    59              08  C0 0013D        ADDL2   #8, HDR_INSERT                          : 0424
```

```
                             68          03  B0 00140         MOVW    #3, (R8)                                   ; 0425
                             60          01  CE 00143         MNEGL   #1, (R0)                                   ; 0426
                       04    A0          01  CE 00146         MNEGL   #1, 4(R0)                                  ; 0427
                             50    02    A6  3C 0014A  12$:   MOVZWL  2(IHD), R0                                 ; 0430
                             2C          50  B1 0014E         CMPW    R0, #44
                                         06  1A 00151         BGTRU   13$
                             59          04  C0 00153         ADDL2   #4, HDR_INSERT                             ; 0433
                             68          04  B0 00156         MOVW    #4, (R8)                                   ; 0434
                                         5A  D4 00159  13$:   CLRL    R10                                        ; 0438
                                         59  D5 0015B         TSTL    HDR_INSERT
                                         17  13 0015D         BEQL    14$
                                         5A  D6 0015F         INCL    R10
                             51          66  3C 00161         MOVZWL  (IHD), R1                                  ; 0443
                             51          50  C2 00164         SUBL2   R0, R1
                57           56          50  C1 00167         ADDL3   R0, IHD, R7                                ; 0444
              6947           67          51  28 0016B         MOVC3   R1, (R7), (HDR_INSERT)[R7]                 ; 0445
      59        00           6E          00  2C 00170         MOVC5   #0, (SP), #0, HDR_INSERT, (R7)             ; 0448
                                         67     00175
                                         58  D4 00176  14$:   CLRL    IHI_INSERT                                 ; 0456
FE78  CF      0C  A6         10          00  ED 00178         CMPZV   #0, #16, 12(IHD), V4_MAJORID               ; 0457
                                         14  1F 00180         BLSSU   15$
FE6E  CF      0C  A6         10          00  ED 00182         CMPZV   #0, #16, 12(IHD), V4_MAJORID               ; 0460
                                         2B  12 0018A         BNEQ    16$
FE68  CF      0E  A6         10          00  ED 0018C         CMPZV   #0, #16, 14(IHD), V4_MINORID               ; 0462
                                         21  1E 00194         BGEQU   16$
                             58          18  D0 00196  15$:   MOVL    #24, IHI_INSERT                            ; 0471
                             50    06    A6  3C 00199         MOVZWL  6(IHD), R0                                 ; 0473
                             51          66  3C 0019D         MOVZWL  (IHD), R1
                             51          50  C2 001A0         SUBL2   R0, R1
                             51          10  C2 001A3         SUBL2   #16, R1
                57           56          50  C1 001A6         ADDL3   R0, IHD, R7                                ; 0474
            28  A7     10    A7          51  28 001AA         MOVC3   R1, 16(R7), 40(R7)                         ; 0475
      58        00           6E          00  2C 001B0         MOVC5   #0, (SP), #0, IHI_INSERT, 16(R7)           ; 0477
                                   10    A7     001B5
                             04          5A  E8 001B7  16$:   BLBS    R10, 17$                                   ; 0484
                                         58  D5 001BA         TSTL    IHI_INSERT                                 ; 0486
                                         29  13 001BC         BEQL    19$
                             50          66  3C 001BE  17$:   MOVZWL  (IHD), R0                                  ; 0489
                             50          59  C0 001C1         ADDL2   HDR_INSERT, R0
                66           50          58  A1 001C4         ADDW3   IHI_INSERT, R0, (IHD)
                       02    A6          59  A0 001C8         ADDW2   HDR_INSERT, 2(IHD)                         ; 0490
                       06    A6          59  A0 001CC         ADDW2   HDR_INSERT, 6(IHD)                         ; 0491
                                   04    A6  B5 001D0         TSTW    4(IHD)                                     ; 0493
                                         04  13 001D3         BEQL    18$
                       04    A6          59  A0 001D5         ADDW2   HDR_INSERT, 4(IHD)                         ; 0495
                             50    08    A6  3C 001D9  18$:   MOVZWL  8(IHD), R0                                 ; 0497
                                         08  13 001DD         BEQL    19$
                             50          59  C0 001DF         ADDL2   HDR_INSERT, R0                             ; 0499
            08  A6           50          58  A1 001E2         ADDW3   IHI_INSERT, R0, 8(IHD)
                             50          01  D0 001E7  19$:   MOVL    #1, R0                                     ; 0503
                                             04 001EA         RET                                                ; 0512

; Routine Size:  491 bytes,    Routine Base:  YF$$SYSIMGACT + 0008


;  430         0513  1
```

```
;  432     0514  1 %SBTTL  'IMG$GET_NEXT_ISD  Get Image Section Descriptor'
;  433     0515  1 GLOBAL ROUTINE IMG$GET_NEXT_ISD
;  434     0516  1   ( CHAN, BLK_BUFADR, IHD_BUFADR, VBN_ADR, OFFSET_ADR, ISD_BUFADR, HEADER_VERSION ) =
;  435     0517  1
;  436     0518  1 !++
;  437     0519  1 ! FUNCTIONAL DESCRIPTION:
;  438     0520  1 !
;  439     0521  1 ! FORMAL PARAMETERS:
;  440     0522  1 !
;  441     0523  1 !       Chan            Channel on which image file is open
;  442     0524  1 !       Blk_bufadr      Address of buffer which contains block of image header
;  443     0525  1 !       Ihd_bufadr      Address of buffer containing decoded IHD
;  444     0526  1 !       VBN_adr         Address of VBN in blk_bufadr
;  445     0527  1 !       Offset_adr      Address of Offset to ISD
;  446     0528  1 !       ISD_bufadr      Address of buffer to contain decoded ISD
;  447     0529  1 !
;  448     0530  1 ! IMPLICIT INPUTS:
;  449     0531  1 !       NONE
;  450     0532  1 !
;  451     0533  1 ! IMPLICIT OUTPUTS:
;  452     0534  1 !       NONE
;  453     0535  1 !
;  454     0536  1 ! ROUTINE VALUE:
;  455     0537  1 ! COMPLETION CODES:
;  456     0538  1 !       NONE
;  457     0539  1 !
;  458     0540  1 ! SIDE EFFECTS:
;  459     0541  1 !       NONE
;  460     0542  1 !
;  461     0543  1 !--
;  462     0544  1
;  463     0545  2 BEGIN
;  464     0546  2 LOCAL
;  465     0547  2     IOSB                : BBLOCK [8],   ! Quadword IO status block
;  466     0548  2     B_ISD               : REF BBLOCK,   ! ISD is header block buffer
;  467     0549  2     ISD                 : REF BBLOCK,
;  468     0550  2     ISD_SIZ             : SIGNED WORD,
;  469     0551  2     SIZE,
;  470     0552  2     STATUS;                             ! Status
;  471     0553  2
;  472     0554  2 BIND
;  473     0555  2     IHD = .IHD_BUFADR           : BBLOCK,
;  474     0556  2     OFFSET = .OFFSET_ADR        : WORD,
;  475     0557  2     VBN = .VBN_ADR;
;  476     0558  2
;  477     0559  2 !
;  478     0560  2 !   Validate that offset and VBN are reasonable
;  479     0561  2 !
;  480     0562  2 IF .OFFSET GEQU
;  481     0563  3         (IF .VBN EQL 1
;  482     0564  3         THEN IMG$C_BLOCKSIZ - 2
;  483     0565  3         ELSE IMG$C_BLOCKSIZ)
;  484     0566  2 THEN
;  485     0567  2     RETURN IMG$_ISD_OFF;
;  486     0568  2
;  487     0569  2 IF .VBN GTR .IHD [IHD$B_HDRBLKCNT]
;  488     0570  2 THEN
```

```
;  489          0571  2      RETURN IMG$_ISD_VBN;
;  490          0572  2
;  491          0573  2  !
;  492          0574  2  !   Get next ISD
;  493          0575  2  !
;  494          0576  2  B_ISD = .BLK_BUFADR + .OFFSET;
;  495          0577  2  ISD_SIZ = .B_ISD [ISD_W_SIZE];
;  496          0578  2
;  497          0579  2  !
;  498          0580  2  !   See whether offset points off the block and we need to read the next block
;  499          0581  2  !
;  500          0582  2  IF .ISD_SIZ EQL -1
;  501          0583  2  THEN                                   ! Read next block
;  502          0584  2      BEGIN
;  503          0585  2      VBN = .VBN + 1;                    ! Increment VBN
;  504          0586  2      OFFSET = 0;
;  505          0587  2      SIZE = IMG$C_BLOCKSIZ;
;  506          0588  2
;  507        P 0589  2      STATUS = $QIOW
;  508        P 0590  2                  (
;  509        P 0591  2                  EFN = EXE$C_SYSEFN,     ! Event flag
;  510        P 0592  2                  CHAN = .CHAN,           ! Channel
;  511        P 0593  2                  FUNC = IO$_READVBLK,    ! Read a virtual block
;  512        P 0594  2                  IOSB = IOSB,            ! I/O status block
;  513        P 0595  2                  P1 = .BLK_BUFADR,       ! Buffer to read in to
;  514        P 0596  2                  P2 = .SIZE,             ! Number of bytes to read
;  515        P 0597  2                  P3 = .VBN               ! Virtual block number to read
;  516          0598  2                  );
;  517          0599  2
;  518          0600  2      IF .STATUS
;  519          0601  2      THEN
;  520          0602  2          STATUS = .IOSB [0,0,16,0];      ! Pick up final status
;  521          0603  2      IF NOT .STATUS
;  522          0604  2      THEN
;  523          0605  2          RETURN .STATUS;
;  524          0606  2
;  525          0607  2      B_ISD = .BLK_BUFADR;
;  526          0608  2      ISD_SIZ = .B_ISD [ISD_W_SIZE];
;  527          0609  2
;  528          0610  2      IF .ISD_SIZ EQL -1                  ! Trap consecutive 'wrap' ISDs
;  529          0611  2      THEN
;  530          0612  2          RETURN IMG$_INCONISD;
;  531          0613  2
;  532          0614  2      END;
;  533          0615  2
;  534          0616  2  !
;  535          0617  2  !   See whether there are any ISDs left
;  536          0618  2  !
;  537          0619  2  IF .ISD_SIZ EQL 0
;  538          0620  2  THEN                                   ! No more ISDs left
;  539          0621  2      RETURN IMG$_ENDOFHDR;
;  540          0622  2
;  541          0623  2  !
;  542          0624  2  !   Validate that the ISD size is reasonable
;  543          0625  2  !
;  544          0626  2  IF (.ISD_SIZ LSS ISD$C_LENDZRO)
;  545          0627  2      OR
```

```
;  546     0628  3      (.ISD_SIZ GTR ISD$C_MAXLENGLBL)
;  547     0629  2  THEN
;  548     0630  2      RETURN IMG$_ISD_SIZ;
;  549     0631  2
;  550     0632  2  !
;  551     0633  2  !   Make sure that ISD doesn't attempt to wrap around to the next block
;  552     0634  2  !
;  553     0635  2  IF (.OFFSET + .ISD_SIZ) GTRU
;  554     0636  2          (IF .VBN EQL 1
;  555     0637  2          THEN IMG$C_BLOCKSIZ - 2
;  556     0638  2          ELSE IMG$C_BLOCKSIZ)
;  557     0639  2  THEN
;  558     0640  2      RETURN IMG$_INCONISD;
;  559     0641  2
;  560     0642  2  ISD = .ISD_BUFADR;
;  561     0643  2  CH$MOVE (.ISD_SIZ, .B_ISD, .ISD);       ! Copy from block to ISD buffer
;  562     0644  2  OFFSET = .OFFSET + .ISD_SIZ;
;  563     0645  2
;  564     0646  2  !
;  565     0647  2  ! Don't use page fault cluster size for cross-linker images
;  566     0648  2  !
;  567     0649  2  IF .HEADER_VERSION EQL IHD$C_GEN_XLNKR
;  568     0650  2  THEN
;  569     0651  2      ISD [ISD$B_PFC] = 0;
;  570     0652  2
;  571     0653  2  !
;  572     0654  2  ! Some V3 images use IHD$L_IAFVA to identify the fixup vectors
;  573     0655  2  !
;  574     0656  2  IF .HEADER_VERSION EQL IHD$C_GEN_FIXUP
;  575     0657  2  THEN
;  576     0658  2      IF (.ISD [ISD$V_VPN] * 512) EQL .IHD [IHD$L_IAFVA]
;  577     0659  2          AND
;  578     0660  2          .ISD [ISD$V_VPN] NEQ 0
;  579     0661  2      THEN
;  580     0662  2          ISD [ISD$V_FIXUPVEC] = 1;
;  581     0663  2
;  582     0664  2  RETURN SS$_NORMAL;
;  583     0665  1  END;                                   ! IMG$GET_NEXT_ISD routine


                                    03FC 00000          .ENTRY  IMG$GET_NEXT_ISD, Save R2,R3,R4,R5,R6,R7,-  ; 0515
                                                                R8,R9
                                5E          08  C2 00002         SUBL2   #8, SP                              ;
                                59      0C  AC  D0 00005         MOVL    IHD_BUFADR, R9                      ; 0555
                                58      14  AC  D0 00009         MOVL    OFFSET_ADR, R8                      ; 0556
                                52      10  AC  D0 0000D         MOVL    VBN_ADR, R2                         ; 0557
                                01          62  D1 00011         CMPL    (R2), #1                            ; 0563
                                            07  12 00014         BNEQ    1$                                  ;
                                50  01FE   8F  3C 00016         MOVZWL  #510, R0                            ; 0564
                                            05  11 0001B         BRB     2$                                  ;
                                50  0200   8F  3C 0001D 1$:      MOVZWL  #512, R0                            ; 0563
                50          68  10          00  ED 00022 2$:      CMPZV   #0, #16, (R8), R0                   ;
                                            08  1F 00027         BLSSU   3$                                  ;
                                50 084D8CB4 8F  D0 00029         MOVL    #139300020, R0                      ; 0567
```

```
                                                 04 00C30                RET
        62      10  A9              08          00 ED 00031 3$:         CMPZV   #0, #8, 16(R9), (R2)                    ;0569
                                                 08 18 00037             BGEQ    4$
                                    50 084D8CBC  8F D0 00039             MOVL    #139300028, R0                          ;0571
                                                 04 00040                RET
                                    53           68 3C 00041 4$:         MOVZWL  (R8), B_ISD                             ;0576
                                    53       08  AC C0 00044             ADDL2   BLK_BUFADR, B_ISD
                                    57           63 B0 00048             MOVW    (B_ISD), ISD_SIZ                        ;0577
                                FFFF    8F       57 B1 0004B             CMPW    ISD_SIZ, #-1                            ;0582
                                                 40 12 00050             BNEQ    7$
                                                 62 D6 00052             INCL    (R2)                                    ;0585
                                                 68 B4 00054             CLRW    (R8)                                    ;0586
                                    50      0200 8F 3C 00056             MOVZWL  #512, SIZE                              ;0587
                                                 7E 7C 0005B             CLRQ    -(SP)                                   ;0598
                                                 7E D4 0005D             CLRL    -(SP)
                                                 62 DD 0005F             PUSHL   (R2)
                                                 50 DD 00061             PUSHL   SIZE
                                            08   AC DD 00063             PUSHL   BLK_BUFADR
                                                 7E 7C 00066             CLRQ    -(SP)
                                            20   AE 9F 00068             PUSHAB  IOSB
                                                 31 DD 0006B             PUSHL   #49
                                            04   AC DD 0006D             PUSHL   CHAN
                                    7E           00G 9A 00070            MOVZBL  S^EXE$C_SYSEFN, -(SP)
                            00000000G   00       0C FB 00073             CALLS   #12, SYS$QIOW
                                    03           50 E9 0007A             BLBC    STATUS, 5$                              ;0600
                                    50           6E 3C 0007D             MOVZWL  IOSB, STATUS                            ;0602
                                    01           50 E8 00080 5$:         BLBS    STATUS, 6$                              ;0603
                                                 04 00083                RET
                                    53       08  AC D0 00084 6$:         MOVL    BLK_BUFADR, B_ISD                       ;0607
                                    57           63 B0 00088             MOVW    (B_ISD), ISD_SIZ                        ;0608
                                FFFF    8F       57 B1 0008B             CMPW    ISD_SIZ, #-1                            ;0610
                                                 3F 13 00090             BEQL    13$
                                                 57 B5 00092 7$:         TSTW    ISD_SIZ                                 ;0619
                                                 08 12 00094             BNEQ    8$
                                    50 084D8640  8F D0 00096             MOVL    #139298368, R0                          ;0621
                                                 04 0009D                RET
                                    0C           57 B1 0009E 8$:         CMPW    ISD_SIZ, #12                            ;0626
                                                 07 19 000A1             BLSS    9$
                                0040    8F       57 B1 000A3             CMPW    ISD_SIZ, #64                            ;0628
                                                 08 15 000A8             BLEQ    10$
                                    50 084D8CC4  8F D0 000AA 9$:         MOVL    #139300036, R0                          ;0630
                                                 04 000B1                RET
                                    51           68 3C 000B2 10$:        MOVZWL  (R8), R1                                ;0635
                                    50           57 32 000B5             CVTWL   ISD_SIZ, R0
                                    51           50 C0 000B8             ADDL2   R0, R1
                                    01           62 D1 000BB             CMPL    (R2), #1                                ;0636
                                                 07 12 000BE             BNEQ    11$
                                    50      01FE 8F 3C 000C0             MOVZWL  #510, R0                                ;0637
                                                 05 11 000C5             BRB     12$
                                    50      0200 8F 3C 000C7 11$:        MOVZWL  #512, R0                                ;0636
                                    50           51 D1 000CC 12$:        CMPL    R1, R0
                                                 08 1B 000CF             BLEQU   14$
                                    50 084D8CAC  8F D0 000D1 13$:        MOVL    #139300012, R0                          ;0640
                                                 04 000D8                RET
                                    56       18  AC D0 000D9 14$:        MOVL    ISD_BUFADR, ISD                         ;0642
                                66  63           57 28 000DD             MOVC3   ISD_SIZ, (B_ISD), (ISD)                 ;0643
                                    68           57 A0 000E1             ADDW2   ISD_SIZ, (R8)                           ;0644
```

```
                                 01        1C    AC  D1 000E4          CMPL    HEADER_VERSION, #1                          ; 0649
                                                 03  12 000E8          BNEQ    15$
                                       07        A6  94 000EA          CLRB    7(ISD)                                      ; 0651
                                 05        1C    AC  D1 000ED 15$:     CMPL    HEADER_VERSION, #5                          ; 0656
                                                 1C  12 000F1          BNEQ    16$
          50     04    A6               15          00  EF 000F3          EXTZV   #0, #21, 4(ISD), R0                         ; 0658
                       50               50          09  78 000F9          ASHL    #9, R0, R0
                                 2C    A9          50  D1 000FD          CMPL    R0, 44(R9)
                                                 0C  12 00101          BNEQ    16$
          00     04    A6               15          00  ED 00103          CMPZV   #0, #21, 4(ISD), #0                         ; 0660
                                                 04  13 00109          BEQL    16$
                                 09    A6          04  88 0010B          BISB2   #4, 9(ISD)                                  ; 0662
                       50               50          01  D0 0010F 16$:     MOVL    #1, R0                                      ; 0664
                                                     04 00112          RET                                                 ; 0665

; Routine Size:  275 bytes,    Routine Base:  YF$$SYSIMGACT + 01F3

;  584          0666  1
```

```
;   586     0667  1 %SBTTL  'CONVERT_XLINK Convert a cross-linker image header to standard format'
;   587     0668  1 ROUTINE CONVERT_XLINK
;   588     0669  1   ( BLK_BUFADR  : REF $BBLOCK,
;   589     0670  1     IHD         : REF $BBLOCK ) =
;   590     0671  1
;   591     0672  1 !++
;   592     0673  1 ! FUNCTIONAL DESCRIPTION:
;   593     0674  1 !       An image header produced by the cross-linker is converted to the
;   594     0675  1 !       standard format.
;   595     0676  1 !
;   596     0677  1 ! FORMAL PARAMETERS:
;   597     0678  1 !
;   598     0679  1 !       Blk_bufadr      Address of buffer which contains first block of image header
;   599     0680  1 !       Ihd             Address of buffer to contain decoded IHD
;   600     0681  1 !
;   601     0682  1 ! IMPLICIT INPUTS:
;   602     0683  1 !       NONE
;   603     0684  1 !
;   604     0685  1 ! IMPLICIT OUTPUTS:
;   605     0686  1 !       NONE
;   606     0687  1 !
;   607     0688  1 ! ROUTINE VALUE:
;   608     0689  1 ! COMPLETION CODES:
;   609     0690  1 !       NONE
;   610     0691  1 !
;   611     0692  1 ! SIDE EFFECTS:
;   612     0693  1 !       NONE
;   613     0694  1 !
;   614     0695  1 !--
;   615     0696  1
;   616     0697  2 BEGIN
;   617     0698  2
;   618     0699  2 BIND
;   619     0700  2     PRIV_MASK = IHD [IHD$Q_PRIVREQS]  : VECTOR [2],
;   620     0701  2     IHD_ACT_ADR = .IHD + IHD$K_LENGTH : VECTOR [3],
;   621     0702  2     IHX_ACT_ADR = BLK_BUFADR [IHX$Q_STARTADR] : VECTOR [2],
;   622     0703  2     IHS         = .IHD + IHD$K_LENGTH + IHA$K_LENGTH : $BBLOCK;
;   623     0704  2 !
;   624     0705  2 ! Zero the one page buffer which will contain decoded IHD
;   625     0706  2 !
;   626     0707  2 CH$FILL (0, 512, .IHD);
;   627     0708  2 !
;   628     0709  2 ! Fill in offsets and directly transportable fields
;   629     0710  2 !
;   630     0711  2 IHD [IHD$W_ACTIVOFF] = IHD$K_LENGTH;
;   631     0712  2 IHD [IHD$W_SIZE] = IHD$K_LENGTH + IHA$K_LENGTH + IHS$K_LENGTH;
;   632     0713  2 IHD [IHD$B_HDRBLKCNT] = .BLK_BUFADR [IHX$B_HDRBLKCNT];
;   633     0714  2 !
;   634     0715  2 ! Convert image ID fields
;   635     0716  2 !
;   636     0717  2 IHD [IHD$W_MAJORID] = IHD$K_MAJORID;
;   637     0718  2 IHD [IHD$W_MINORID] = IHD$K_MINORID;
;   638     0719  2 !
;   639     0720  2 ! Assume all privileges
;   640     0721  2 !
;   641     0722  2 PRIV_MASK [0]= -1;
;   642     0723  2 PRIV_MASK [1]= -1;
```

```
;  643      0724  2 !
;  644      0725  2 ! Add image activation data
;  645      0726  2 !
;  646      0727  2 IHD_ACT_ADR [0] = .IHX_ACT_ADR [0];
;  647      0728  2 IHD_ACT_ADR [1] = .IHX_ACT_ADR [1];
;  648      0729  2 !
;  649      0730  2 ! Check for DEBUG data
;  650      0731  2 !
;  651      0732  2 IF .BLK_BUFADR [IHX$W_MINORID] GEQ IHX$K_MINORID1
;  652      0733  2 THEN
;  653      0734  3     BEGIN
;  654      0735  3     IHD [IHD$W_SYMDBGOFF] = IHD$K_LENGTH + IHA$K_LENGTH;
;  655      0736  3     IHD_ACT_ADR [2] = .BLK_BUFADR [IHX$L_TFRADR3];
;  656      0737  3     IHS [IHS$L_DSTVBN] = .BLK_BUFADR [IHX$L_DSTVBN] ;
;  657      0738  3     IHS [IHS$L_GSTVBN] = .BLK_BUFADR [IHX$L_GSTVBN] ;
;  658      0739  3     IHS [IHS$W_DSTBLKS] = .BLK_BUFADR [IHX$W_DSTBLKS] ;
;  659      0740  3     IHS [IHS$W_GSTRECS] = .BLK_BUFADR [IHX$W_GSTRECS] ;
;  660      0741  2     END;
;  661      0742  2
;  662      0743  2 RETURN SS$_NORMAL;
;  663      0744  1 END;


                                         0FFC 00000 CONVERT_XLINK:
                                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11              ; 0668
                        56        08  AC  D0 00002        MOVL    IHD, R6                                           ; 0700
                        5B        14  A6  9E 00006        MOVAB   20(R6), R11                                       ;
                        58        30  A6  9E 0000A        MOVAB   48(R6), R8                                        ; 0701
                        57        04  AC  D0 0000E        MOVL    BLK_BUFADR, R7                                    ; 0702
                        5A        04  A7  9E 00012        MOVAB   4(R7), R10                                        ;
                        59        44  A6  9E 00016        MOVAB   68(R6), R9                                        ; 0703
   0200   8F        00  6E            00  2C 0001A        MOVC5   #0, (SP), #0, #512, (R6)                          ; 0707
                                      66     00021
                     66 00300058      8F  D0 00022        MOVL    #3145816, (R6)                                    ; 0712
                  10 A6       02      A7  90 00029        MOVB    2(R7), 16(R6)                                     ; 0713
                  0C A6 35303230      8F  D0 0002E        MOVL    #892351024, 12(R6)                                ; 0717
                     6B               01  CE 00036        MNEGL   #1, (R11)                                         ; 0722
                  04 AB               01  CE 00039        MNEGL   #1, 4(R11)                                        ; 0723
                     68               6A  7D 0003D        MOVQ    (R10), (R8)                                       ; 0727
                3130 8F       0E      A7  B1 00040        CMPW    14(R7), #12592                                    ; 0732
                                      13  1F 00046        BLSSU   1$                                                ;
                  04 A6       44      8F  9B 00048        MOVZBW  #68, 4(R6)                                        ; 0735
                  08 A8       34      A7  D0 0004D        MOVL    52(R7), 8(R8)                                     ; 0736
                     69       28      A7  7D 00052        MOVQ    40(R7), (R9)                                      ; 0737
                  08 A9       30      A7  D0 00056        MOVL    48(R7), 8(R9)                                     ; 0739
                     50               01  D0 0005B 1$:    MOVL    #1, R0                                            ; 0743
                                          04 0005E        RET                                                       ; 0744

; Routine Size:  95 bytes,    Routine Base:  YF$$SYSIMGACT + 0306
```

```
;  665          0745  1 END                                   !End of module IMGDECODE
;  666          0746  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  YF$$SYSIMGACT                    869  NOVEC,  WRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                                 Library Statistics
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619        63         0        1000        00:01.8
```

```
;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:IMGDECODE/OBJ=OBJ$:IMGDECODE MSRC$:IMGDECODE/UPDATE=(ENH$:IMGDECODE)

; Size:          861 code + 8 data bytes
; Run Time:          00:19.1
; Elapsed Time:      00:22.0
; Lines/CPU Min:      2348
; Lexemes/CPU-Min:  16064
; Memory Used:   205 pages
; Compilation Complete
```